

AMENDMENTS TO THE CLAIMS

1. (Original) A method for tracking kernel resource usage comprising the steps of:
generating a tag to charge a process allocated with kernel resources;
determining whether the process is a kernel process or a user process; and,
flagging the tag to indicate whether the process is a kernel process or a user process
based upon the determination step.

2. (Original) The method of claim 1 wherein said step of generating a tag further
comprises the steps of:
determining whether a request for kernel resources is passed from an intermediate
function using a worker thread;
finding at least one link between the worker thread and the process; and,
identifying the process that originated the request according to the found link between
the worker thread and the process.

3. (Original) The method of claim 1 wherein said step of flagging the tag further
comprises the steps of:
generating a tag value to identify the kernel resources allocated to a kernel process;
and,
saving a driver identification to the tag value.

4. (Original) The method of claim 3 wherein the tag value with the driver
identification is saved in a first word of the tag.

5. (Original) The method of claim 1 wherein said step of flagging the tag further
comprises the steps of:
generating a tag value to identify kernel resources allocated to a user process;
saving a type of kernel resources allocated to the tag; and,
saving a user process identifier to the tag to identify the process.

6. (Original) The method of claim 5 wherein the tag value and the type of kernel resources are saved in a first word of the tag, and the user process identifier is saved in a second word of the tag.

7. (Original) The method of claim 1 wherein said step of flagging the tag further comprises the steps of:

generating a tag value to identify kernel resources allocated to a user process;
saving the tag value to a first word of the tag; and,
saving a user process identifier to identify the process to a second word of the tag.

8. (Original) A method for tracking kernel resource usage comprising the steps of:
generating a tag to charge a process allocated with kernel resources;
determining whether the process is a first predefined process or a second predefined process; and,

saving an identifier to the tag to identify whether the process is a first predefined process or a second predefined process based upon the determination step.

9. (Original) The method of claim 8 further comprising the steps of:
saving a process identifier to identify the process allocated with the kernel resources;
and,
saving a type of kernel resources allocated to the process.

10. (Original) The method of claim 8 wherein said step of generating a tag further comprises the steps of:

determining whether a request for kernel resources is passed from an intermediate function using a worker thread;
finding at least one link between the worker thread and the process; and,
identifying the process that originated the request according to the found link between the worker thread and the process.

11. (Original) The method of claim 8 wherein said step of saving an identifier further comprises the step of flagging the tag as the first predefined process or the second predefined process according to the determination step.

12. (Original) The method of claim 8 wherein the first and second predefined processes relate to processes from kernel level or user level in an operating system.

13. (Original) The method of claim 8 wherein the first and second predefined processes relate to processes used by users, a group of users, or accounts of the users in a network system.

14. (Original) A method for tracking kernel resource usage comprising the steps of:
generating a tag to charge a process called from user level of an operation system allocated with kernel resources;

saving a tag value to the tag to identify the kernel resources allocated to the user process; and,

saving a user process identifier to the tag to identify the user process.

15. (Original) The method of claim 14 wherein the tag value is saved in a first word of the tag, and the user process identifier is saved in a second word of the tag.

16. (Original) The method of claim 14 further comprising the step of flagging the tag as a process called from the user level of an operating system.

17. (Original) The method of claim 14 wherein said step of saving a tag value further comprises the steps of:

determining whether a request for kernel resources is passed from an intermediate function using a worker thread;

finding at least one link between the worker thread and the process; and,

identifying the process that originated the request according to the found link between the worker thread and the process.

18. (Original) The method of claim 14 further comprising the step of saving a type of kernel resources allocated to the user process to the tag.

19. (Original) The method of claim 14 wherein said step of saving a user process identifier further comprises the steps of:

extending the tag with a second word; and,
saving the user process identifier to the second word of the tag.

20. (Original) The method of claim 19 further comprising the step of saving the tag value to a first word of the tag.

21. (Original) A method for tracking kernel resources allocated to kernel and user processes indicated by a plurality of tags, the method comprising the steps of:

identifying an amount of kernel resources allocated to a process indicated by a selected tag;

saving an association of the identified amount of kernel resources allocated with the process to a file;

identifying a type of the kernel resources allocated to the process indicated by the selected tag; and,

saving an association of the identified type of the kernel resources allocated with the process to the file.

22. (Original) The method according to claim 21 further comprising the steps of:
selecting a tag from the plurality of tags; and,
saving the selected tag to the file.

23. (Original) The method according to claim 21 further comprising the steps of:
determining whether there are any more tags from the plurality of tags; and,
repeating the method for any other tags based upon the determination step.

24. (Original) The method according to claim 21 wherein the step of identifying the type of kernel resources further comprises the steps of:

determining whether the process is called from the user level; and,

performing the step of identifying the type of kernel resources when the process is called from the user level.

25. (Original) A method for managing kernel resource usage comprising the steps of:

reading a usage policy, wherein the usage policy includes data relating to threshold limits according to processes and kernel resource type;

searching a file to determine an amount of kernel resources used by each process, wherein the file includes a plurality of tags having a tag value identifying the kernel resources allocated to the process, a process identifier identifying the process allocated with the kernel resources, a flag identifying whether the process is a first predefined process or a second predefined process, and a kernel resource type identifying the type of kernel resources allocated to the process;

determining whether the amount of kernel resource usage exceeds the threshold limits according to the user policy; and,

taking an action according to the usage policy when the amount of kernel resource usage is over the threshold limits.

26. (Original) The method according to claim 25 further comprising the steps of: determining whether the process should be aborted according to the usage policy; aborting the process according to the determination step; and, updating the file to reflect the aborted process.

27. (Currently amended) A tangible computer-readable storage medium having stored thereon a data structure, comprising:

a first field containing a user process identifier to identify a user process allocated with kernel resources; and,

a second field containing a value to identify kernel resources allocated to the user process.

28. (Original) The medium of claim 27 further comprising a third field containing a flag that indicates the user process as a process called from user level.

29. (Original) The medium of claim 27 wherein the first field is a first long word of a tag, and the second field is a second long word of the tag.

30. (Currently amended) A tangible computer-readable storage medium having computer-executable instructions for performing steps comprising:

generating a tag to charge a process allocated with kernel resources;
determining whether the process is a kernel process or a user process; and,
flagging the tag to indicate whether the process is a kernel process or a user process
based upon the determination step.

31. (Currently amended) A tangible computer-readable storage medium having
computer-executable instructions for performing steps comprising:
generating a tag to charge a process allocated with kernel resources;
determining whether the process is a first predefined process or a second predefined
process; and,
saving an identifier to the tag to identify whether the process is a first predefined
process or a second predefined process based upon the determination step.

32. (Currently amended) A tangible computer-readable storage medium having
computer-executable instructions for performing steps comprising:
generating a tag to charge a process called from user level of an operating system
allocated with kernel resources;
saving a tag value to the tag to identify the kernel resources allocated to the user
process; and,
saving a user process identifier to the tag to identify the user process.

33. (Currently amended) A tangible computer-readable storage medium having
computer-executable instructions for performing steps comprising:
identifying an amount of kernel resources allocated to a process indicated by a
selected tag;
saving an association of the identified amount of kernel resources allocated with the
process to a file;
identifying a type of the kernel resources allocated to the process; and,
saving an association of the identified type of the kernel resources allocated with the
process to the file.

34. (Currently amended) A tangible computer-readable storage medium having computer-executable instructions for performing steps comprising:

- reading a usage policy, wherein the usage policy includes data relating to threshold limits according to processes and kernel resource type;
- searching a file to determine an amount of kernel resources used by each process, wherein the file includes a plurality of tags having a tag value identifying the kernel resources allocated to the process, a process identifier identifying the process allocated with the kernel resources, a flag identifying whether the process is a first defined process or a second defined process, and a kernel resource type identifying the type of kernel resources allocated to the process;
- determining whether the amount of kernel resource usage exceeds the threshold limits according to the user policy; and,
- taking an action according to the usage policy when the amount of kernel resource usage is over the threshold limits.